

## まえがき

本書は、プログラミングやコンピュータに興味を持つ者が、コンパイラの原理と構造、さらにその開発方法を理解するための教科書である。

コンパイラの理解は、コンピュータの原理やプログラムの実行のしくみを理解する基礎である。さらに、長年のコンパイラの研究開発を通じて築きあげられた理論や技術には、情報システムにおける問題解決の典型的な例が多く含まれている。情報系のカリキュラムでコンパイラを学ぶ意義の一つは、その背後にあるアイデアを理解し、理論体系やシステム構築の過程を追体験することといえる。これらの体験は、情報分野の研究や開発に従事する上での貴重な財産となるはずである。そこで本書では、コンパイラの役割とその構造を体系的に理解したあと、コンパイラ実現のための主要な基盤技術を、その原理とその背後にある考え方を含め習得することを目標とする。この目標を念頭に、従来のコンパイラの解説とは異なる以下のような構成をとる。

まず第1章で、Turing らによって確立された計算可能性の理論と万能計算機の構築方法を基礎として、コンパイラを含むプログラミング言語処理系の構築原理とその構造を理解する。体系的な理解を目標とするが、計算の理論を初めて学ぶ者も理解できるように、例を交えて具体的に解説する。この理解を基礎に、第3章から第6章までの各章で、プログラミング言語を、文字列集合から型の関係に至る階層的な構造と捉え、各層について、その定義と

解析方法を学ぶ。これらの各章では、定義階層ごとに重要な技術に焦点を当て、技術の使い方のみならず、それらの技術が基礎とする原理とその原理の背後にある考え方を含む系統的な理解につながる解説を試みる。考え方を理解できれば、従来難解と見なされている LR 構文解析や多相型の型推論などの技術も見通しよく習得できるはずである。言語の解析方法を習得したあと、第 7 章で、再帰関数や高階の関数を含むプログラムの操作的意味とインタープリタの構成方法を学ぶ。第 8 章では、機械語コードの生成方法を理解するために、抽象機械を定義し、高水準言語を抽象機械コードにコンパイルする系統的な方法とその正しさの証明手法を学ぶ。

以上のコンパイラの原理と構造の理解に加えて、先端的言語コンパイラの開発方法を習得するために、実際に実行可能なインタープリタと抽象機械へのコンパイラを開発する。高階の関数や多相型型推論などの先端機能を含む ML 言語の小さなサブセット CoreML を対象とし、ML 言語の一つである SML# を用いてその処理系を開発する。第 2 章で、その準備として、第 1 章で学んだチューリング機械のインタープリタを題材とし SML# プログラミングの基本を学び、それに続く各章で、コンパイラの各要素技術の理解と並行して、CoreML 処理系を段階的に開発する。

本書は、筆者が東北大学で長年担当してきた工学部専門科目「コンパイラ」の講義資料に基づく。巻末の参考文献で、より深く学ぶための論文や教科書、本書を執筆するにあたって参考にした文献などを紹介する。また、本書を学習する上で参考になる情報やデータを掲載したサポートページを

<https://AtsushiOtori.github.io/ja/texts/compiler/>

に開設している（AtsushiOtori の大文字と小文字は区別されない）。

本書の出版に際してお世話になった、共立出版株式会社の影山綾乃氏はじめ担当者の方々に深謝する。

2021 年 8 月

大堀 淳